Stat Comp HW#3 Jason Rights

Due Thursday, 08 October, 1:00 PM

50 points total.

$5^{n=day}$ points taken off for each day late.

This assigment includes turning in the first two assignments. All three should include knitr files (named `homework1.rmd`, `homework2.rmd`, `homework3.rmd`) along with valid PDF output files. Inside each file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to properly name files or include author name may result in 5 points taken off.

Question 1

10 points

1. Use GitHub to turn in the first three homework assignments. Make sure the teacher (couthcommander) and TA (trippcm) are collaborators. (5 points)
2. Commit each assignment individually. This means your repository should have at least three commits. (5 points)

Question 2

15 points

Write a simulation to calculate the power for the following study design. The study has two variables, treatment group and outcome. There are two treatment groups (0, 1) and they should be assigned randomly with equal probability. The outcome should be a random normal variable with a mean of 60 and standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome. 5 is the true treatment effect. Create a linear of model for the outcome by the treatment group, and extract the p-value (hint: see assigment1). Test if the p-value is less than or equal to the alpha level, which should be set to 0.05.

```r
set.seed(12345)
gen.data <- function(N){
  treatment <- rbinom(N,size=1,prob=.5)
  outcome <- rnorm(N,60,20)
  suppressWarnings(dat <- as.data.frame(cbind(treatment,outcome)))
  for(i in seq(N)){
     outcome[i] <- ifelse(treatment[i] == 1, outcome[i] + 5, outcome[i])
  }
  #suppressWarnings(attach(dat))
  dat <- cbind()
  dat.mod <- lm(outcome ~ treatment)
  p.value <- coef(summary(lm(dat.mod)))
  return(p.value[2,4])
}
gen.data(100)
```

```
## [1] 0.2881805
```

The p-value for this trial is not less than .05.

Repeat this procedure 1000 times. The power is calculated by finding the percentage of times the p-value is less than or equal to the alpha level. Use the `set.seed` command so that the professor can reproduce your results.

```
set.seed(12345)
invisible(x <- replicate(1000,gen.data(100)))
length(x[which(x<.05)])/1000
```

```
## [1] 0.225
```

1. Find the power when the sample size is 100 patients. (10 points)

```
set.seed(12345)
invisible(x <- replicate(1000,gen.data(100)))
length(x[which(x<.05)])/1000
```

```
## [1] 0.225
```

1. Find the power when the sample size is 1000 patients. (5 points)

```
set.seed(12345)
invisible(x <- replicate(1000,gen.data(1000)))
length(x[which(x<.05)])/1000
```

```
## [1] 0.973
```

Question 3

15 points

Obtain a copy of the football-values lecture. Save the `2015/proj_rb15.csv` file in your working directory. Read in the data set and remove the first two columns.

```
football.rb <- read.csv("https://raw.githubusercontent.com/couthcommander/football-values/a19eb6f5f6862
write.table(football.rb,"football.rb.csv")
```

1. Show the correlation matrix of this data set. (3 points)

```
cor(football.rb[,3:8])
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
```

1. Generate a data set with 30 rows that has a similar correlation structure. Repeat the procedure 10,000 times and return the mean correlation matrix. (10 points)

```r
library(MASS)
mvrnorm(30, mu=colMeans(football.rb[,3:8]), Sigma = cor(football.rb[,3:8]))
```

```
##       rush_att rush_yds  rush_tds  rec_att  rec_yds      rec_tds
##  [1,] 62.80255 270.6337 0.9453858 14.32232 115.0275  0.002423347
##  [2,] 64.03133 271.9122 2.0511578 15.84949 116.5476  1.240765813
##  [3,] 63.22618 271.2088 1.8646727 14.60794 115.1918  0.208863160
##  [4,] 62.37876 270.3007 0.5555703 13.04274 113.7018 -0.403438869
##  [5,] 62.83902 270.6841 0.8026576 14.09899 114.9431  1.009202426
##  [6,] 64.13901 272.0319 2.0996282 16.58215 117.3447  2.881530378
##  [7,] 63.67781 271.5324 1.6930056 14.32223 115.0187 -0.631613762
##  [8,] 62.53794 270.4676 1.0151146 13.63849 114.4397 -0.190443623
##  [9,] 63.11785 271.0936 1.6476927 12.78642 113.6326  0.400531583
## [10,] 62.90060 270.8395 1.5633128 13.48158 114.2853  0.248473358
## [11,] 64.17269 271.9560 2.5621815 15.07863 115.6324  0.684318197
## [12,] 64.78739 272.7774 3.4172749 15.79063 116.4257  2.121468932
## [13,] 63.55413 271.4342 1.8254477 14.24065 114.9913  0.265115137
## [14,] 64.29285 272.2044 2.6605055 16.79362 117.5855  2.293064576
## [15,] 62.40899 270.3155 0.5880161 14.04481 114.8430  0.269788856
## [16,] 64.28423 272.2656 2.6412715 14.49536 115.0905  0.359074600
## [17,] 62.29470 270.2377 0.6169480 14.74299 115.5263  0.325898925
## [18,] 62.75410 270.6350 1.1615009 13.77235 114.6236  0.447498452
## [19,] 63.91198 271.8430 2.3043662 15.69974 116.4790  2.503266969
## [20,] 62.82331 270.8331 1.3208037 13.10497 113.9221  0.437391136
## [21,] 63.44896 271.3412 2.0504757 14.10350 114.7891  0.943434983
## [22,] 63.28038 271.1632 1.3539279 14.00771 114.7654 -0.697339472
## [23,] 62.10344 270.0837 0.5763361 13.25135 114.0026 -0.543673631
## [24,] 65.34642 273.3125 3.5234148 15.98496 116.6258  1.858332875
## [25,] 63.23177 271.2186 1.6543784 12.27875 113.0634 -0.937559675
## [26,] 63.79450 271.6745 2.1508255 15.45266 116.2302  2.344828139
## [27,] 63.85445 271.7797 2.2455299 15.22090 115.8734  0.727610956
## [28,] 63.96318 271.9041 2.3607263 15.04836 115.7227  1.655615411
## [29,] 63.62534 271.5704 2.0124796 14.24389 114.9747 -0.189263248
## [30,] 63.09227 271.1092 2.3060721 14.24855 115.0669  0.689659314
```

```r
x <- replicate(10000,mvrnorm(30, mu=colMeans(football.rb[,3:8]), Sigma = cor(football.rb[,3:8])))
rowMeans(colMeans(x))
```

```
##    rush_att    rush_yds    rush_tds     rec_att     rec_yds     rec_tds
##  63.4644817 271.3940724   1.8320434  14.4664492 115.1810894   0.5388675
```

```r
keep.1 <- 0
loops <- 10000
for(i in seq(loops)){
    keep.1 <- keep.1 + cor(x[,,i])/loops
}
keep.1
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974529 0.9714862 0.7640209 0.7345557 0.5914357
## rush_yds 0.9974529 1.0000000 0.9767845 0.7592335 0.7288946 0.5967383
## rush_tds 0.9714862 0.9767845 1.0000000 0.7209078 0.6928495 0.5857691
```

```
## rec_att  0.7640209 0.7592335 0.7209078 1.0000000 0.9942009 0.8342263
## rec_yds  0.7345557 0.7288946 0.6928495 0.9942009 1.0000000 0.8477534
## rec_tds  0.5914357 0.5967383 0.5857691 0.8342263 0.8477534 1.0000000
```

1. Generate a data set with 30 rows that has the exact correlation structure as the original data set. (2 points)

```r
x <- mvrnorm(30, mu=colMeans(football.rb[,3:8]), Sigma = cor(football.rb[,3:8]),empirical=TRUE)
cor(x)
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
```

Question 4

10 points

Use LaTeXto create the following expressions.

1.

$$P(B) = \sum_j P(B|A_j)P(A_j),$$

$$\Rightarrow P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j (B|A_j)P(A_j)}$$

2.

$$\hat{f}(\zeta) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x \zeta} dx$$

3.

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left[\tfrac{\partial \mathbf{f}}{\partial x_1} \cdots \tfrac{\partial \mathbf{f}}{\partial x_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$